Simulation-Optimization via Kriging and Bootstrapping

Kleijnen, Jack P.C.

Publication date:
2013

No. 2013-064

SIMULATION-OPTIMIZATION VIA KRIGING AND BOOTSTRAPPING: A SURVEY

By

Jack P.C. Kleijnen

1 November, 2013

This is a revised version of CentER Discussion Paper

No. 2011-064
6 June, 2011

ISSN 0924-7815
ISSN 2213-9532

Simulation-optimization via Kriging and bootstrapping: a survey

Jack P.C. Kleijnen
Department of Information Management / CentER,
Tilburg University, Postbox 90153, 5000 LE Tilburg,
Netherlands, phone +31-13-466-2029; fax +31-13-466-3069;
kleijnen@tilburguniversity.edu

Abstract

This article surveys optimization of simulated systems. The simulation may be either deterministic or random. The survey reflects the author's extensive experience with simulation-optimization through Kriging (or Gaussian process) metamodels. The analysis of these metamodels may use parametric bootstrapping for deterministic simulation or distribution-free bootstrapping (or resampling) for random simulation. The survey covers: (1) Simulation-optimization through "efficient global optimization" (EGO) using "expected improvement" (EI); this EI uses the Kriging predictor variance, which can be estimated through parametric bootstrapping accounting for estimation of the Kriging parameters. (2) Optimization with constraints for multiple random simulation outputs and deterministic inputs through mathematical programming applied to Kriging metamodels validated through distribution-free bootstrapping. (3) Taguchian robust optimization for uncertain environments, using mathematical programming—applied to Kriging metamodels—and distribution-free bootstrapping to estimate the variability of the Kriging metamodels and the resulting robust solution. (4) Bootstrapping for improving convexity or preserving monotonicity of the Kriging metamodel.

Keywords: simulation; optimization; stochastic process; nonlinear programming; risk

JEL: C0, C1, C9

1 Introduction

In this survey we consider the problem of optimizing complex real-life systems that are represented through simulation models. These simulation models may be either deterministic or random. *Deterministic* models often represent real-life systems that are governed by laws of physics;

examples are found in computer aided engineering; see [16] and the references in [23, p. 3]. *Random* or stochastic simulation models—including discrete-event simulation—often represent social systems in which humans create noise; examples are queueing systems in telecommunications and logistics with random customer arrival and service times; see [32]. Finding the optimal input combinations for these simulation models may use a large variety of methods; see [19] and [23]. Notice that "input combinations" of a simulation are also called "points" in the search space or "scenarios" that users wish to explore.

We focus our survey on our own research—performed together with various coauthors—on simulation-optimization through Kriging metamodels. In the simulation literature, a *metamodel* is an explicit simplified model of an underlying complicated simulation model; metamodels are also called approximations, response surfaces, surrogates, emulators, etc. Though the most popular type of metamodel is either a first-order polynomial or a second-order polynomial with the simulation inputs as independent variables, we focus on *Kriging* metamodels. The mathematical foundation of Kriging —originally proposed by the South African mining engineer Krige—is due to the French mathematician Matheron, who considered Kriging models as Gaussian process (GP) models; one of his English publications is [39].

For the analysis of Kriging metamodels we use *bootstrapping*. In general, bootstrapping is a versatile method for analyzing nonlinear statistics; an example of a nonlinear statistic is the ratio of two random variables (say) x/y, for which it is well-known that $E(x/y) \neq E(x)/E(y)$. A more interesting example is the variance of the predictor given by a Kriging metamodel with estimated parameters; see Section 4. We shall discuss both parametric bootstrapping for deterministic simulation and nonparametric or distribution-free bootstrapping for random simulation with replications. The bootstrap method avoids complicated asymptotic methods; i.e., bootstrapping is a practical small-sample method—and small samples are common in so-called expensive simulation, which requires much computer time.

We shall also mention *software* for simulation-optimization, Kriging, and bootstrapping; obviously, such software stimulates the application of methods in practice. We use only basic mathematics and statistics in our survey of these various methods. Our 55 references enable readers to study the technical details of these methods. We use a frequentist approach, not a Bayesian approach; the latter is also popular in Kriging and simulation-optimization, but we have no personal experience with Bayesian methods.

To situate our own research within the general context of simulation-

optimization, we give several references at the start of the various sections. Other references enable the readers to learn the details of our own methods. Future research per topic is also mentioned in the various sections.

2 Kriging: basics

Originally, Kriging was developed by Daniel Krige for the interpolation of geostatistical (spatial) sampling data; see [9]. Later on, Kriging was applied to obtain a global (not local) metamodel for the input/output (I/O) data of computer experiments with deterministic simulation models; see [48] and also [49]) and [16]. A recent survey of Kriging in simulation-based metamodeling is [24]. The literature on Kriging is vast and covers diverse disciplines, such as spatial statistics, machine learning, mechanical engineering, and operations research. Besides the Anglo-Saxon literature there is a vast French literature on Kriging; see `http://www.gdr-mascotnum.fr/documents.html`.

The following website emphasizes machine learning: `http://www.gaussianprocess.org/`. This site also gives alternative Kriging books such as [45] and [51].

There is much *software* for Kriging; see the preceding textbooks and websites. In all our own experiments, however, we have used DACE, which is a free-of-charge Matlab-toolbox well documented in [35]. Alternative free software is mentioned in [17] and [23, p. 146]; also see the toolboxes called Surrogates and SUMO on `http://sites.google.com/site/felipeacviana/surrogatestoolbox` and `http://www.sumo.intec.ugent.be/?q=sumo_toolbox`.

The statistical R community has also developed much software; see, e.g., [10]'s mlegp and [47]'s DiceKriging. Some publications focus on problems caused by large I/O data sets (so the matrix inversions in the equations 3 and 4 below become problematic); also see the topic "approximations" on `http://www.gaussianprocess.org/#approx`.

Kriging may give a valid metamodel (i.e., an adequate explicit approximation) of the implicit I/O function implied by the underlying simulation model—even when the simulation experiment covers a "big" input area so the simulation experiment is *global* (not local). For example, Kriging can approximate the I/O function of a simulation model for a traffic rate ranging all the way between (say) 0.1 and 0.9.

Ordinary Kriging—simply called "Kriging" in the remainder of this paper—assumes that the I/O function being approximated is a realization of the GP

$$Y(\mathbf{x}) = \mu + Z(\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is a point in a d-dimensional input space with d a given positive

integer denoting the number of simulation input variables, μ is its constant mean, and $Z(\mathbf{x})$ is a stationary GP with mean zero, variance σ^2, and some correlation function. The most popular correlation function in simulation is the product of the individual correlation functions; e.g.

$$corr[Y(\mathbf{x}_i),Y(\mathbf{x}_j)] = \prod_{k=1}^{d} \exp(-\theta_k \, |x_{ik} - x_{jk}|^{p_k}) \text{ with } \theta_k > 0, p_k. \quad (2)$$

This function implies that the outputs $Y(\mathbf{x}_i)$ and $Y(\mathbf{x}_j)$ are more correlated as their input locations $\mathbf{x}_i$ and $\mathbf{x}_j$ are "closer"; i.e., they have smaller Euclidean distance in the k^{th} dimension of the input combinations $\mathbf{x}_i$ and $\mathbf{x}_j$. The correlation parameter θ_k denotes the importance of input k; i.e., the higher θ_k is, the faster the correlation function decreases with the distance in this dimension. The parameter p_k determines the smoothness of the correlation function; e.g., $p_k = 2$ gives the so-called *Gaussian correlation function*, which gives smooth, continuous functions that are infinitely differentiable. Hence, (1) and (2) imply that the Kriging *(hyper)parameters* are $\psi' = (\mu, \sigma^2, \theta')$ with $\theta' = (\theta_1, \ldots \theta_d)$.

The Kriging predictor is selected such that it is the *best linear unbiased predictor* (BLUP) where "best" means minimal "mean squared error" (MSE) of the predictor. Given a set of n "old" observations (or "training points" in the terminology of machine learning) $\mathbf{y} = (y_1, \ldots, y_n)'$, it can be proven that this criterion gives the following *linear* predictor for a point $\mathbf{x}_{n+1}$ (sometimes denoted by $\mathbf{x}_0$), which may be either a new or an old point:

$$\widehat{y}(\mathbf{x}_{n+1}) = \mu + \mathbf{r}'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu) \quad (3)$$

where $\mathbf{r} = \{corr[Y(\mathbf{x}_{n+1}), Y(\mathbf{x}_1)], \ldots, corr[Y(\mathbf{x}_{n+1}), Y(\mathbf{x}_n)]\}'$ is the vector of correlations between the outputs at the point $\mathbf{x}_{n+1}$ and the n old points $\mathbf{x}_i$, $\mathbf{R}$ is the $n \times n$ matrix with entry (i, j) given by (2), and $\mathbf{1}$ is the n-dimensional vector with ones. This $\mathbf{r}$ and $\mathbf{R}$ may be replaced by the corresponding covariance vector $\mathbf{\Sigma}_{n+1} = \sigma^2\mathbf{r}$ and matrix $\mathbf{\Sigma} = \sigma^2\mathbf{R}$, because σ^2 then cancels out in (3). If the new $\mathbf{x}_{n+1}$ coincides with an old $\mathbf{x}_i$, then the predictor $\widehat{y}(\mathbf{x}_i)$ equals the observed value $y(\mathbf{x}_i)$: *exact interpolation*. Kriging tends to give a bad predictor in case of extrapolation, so we avoid extrapolation; see [2, p. 9].

A major problem is that the Kriging parameters ψ are unknown. To estimate ψ, the standard literature and software use maximum likelihood estimators (MLEs). The MLE of ψ is $\widehat{\psi} = (\widehat{\mu}, \widehat{\sigma^2}, \widehat{\theta'})'$. Computing $\widehat{\psi}$ involves constrained maximization, which is a hard problem because matrix inversion is necessary, the likelihood function may have multiple local maxima and a ridge, etc.; see [38]. It turns out that $\widehat{\mu}$ is the generalized least squares (GLS) estimator $\widehat{\mu} = (\mathbf{1}^T\widehat{\mathbf{R}}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\widehat{\mathbf{R}}^{-1}\mathbf{y}$ where

$\widehat{\mathbf{R}}$ denotes the MLE of $\mathbf{R}$ (determined by θ). [16, p. 84] proves that the MSE of the BLUP or the *predictor variance* is

$$\sigma^2(\mathbf{x}) = \sigma^2(1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}) \tag{4}$$

where $\sigma^2(\mathbf{x})$ is the variance of $\widehat{y}(\mathbf{x})$, defined in (3).

The classic Kriging literature, software, and practice simply replace $\mathbf{R}$ and $\mathbf{r}$ in (3) and (4) by their estimators. Unfortunately, *plugging-in* these estimators into (3) changes $\widehat{y}(\mathbf{x})$ into the *nonlinear* predictor (say) $\widehat{\widehat{y}}(\mathbf{x})$ (double hats). The classic literature ignores this complication, and simply plugs $\widehat{\sigma^2}$, $\widehat{\mathbf{r}}$, and $\widehat{\mathbf{R}}$ into (4) to obtain (say) $s^2(\mathbf{x})$, the *estimated predictor variance* of $\widehat{\widehat{y}}(\mathbf{x})$. There is abundant software for the computation of $\widehat{\widehat{y}}(\mathbf{x})$ and $s^2(\mathbf{x})$. Obviously, $s^2(\mathbf{x}) = 0$ at the n old points; $s^2(\mathbf{x})$ tends to increase as the new $\mathbf{x}_{n+1}$ lies farther away from an old $\mathbf{x}_i$ (also see Sections 4 and 5).

Obviously, the interpolation property of Kriging is not desirable in *random* simulation. Therefore the Kriging metamodel may be changed such that it includes *intrinsic* noise; see (in historical order) [49, pp. 215-249], [16, p. 143], [55], [1], and [6]. The resulting "stochastic Kriging" does not interpolate the n outputs averaged over the (say) m_i replications ($i = 1, \ldots, n$). Moreover, [6] also accounts for common random numbers (CRN), used to simulate outputs for different input combinations (also see later sections). This stochastic Kriging may avoid overfitting; overfitting may result in a wiggling (erratic) Kriging metamodel (also see Section 8). ([37] use an alternative approach with one Kriging model for the mean and another Kriging model for the variance.) More specifically, stochastic Kriging augments (1) with a *white noise* random variable (say) e:

$$Y(\mathbf{x}) = \mu + Z(\mathbf{x}) + e \tag{5}$$

where e is normally, independently, and identically distributed (NIID) with zero mean and (constant) variance σ_e^2. This e is generalized in [1] such that e has a variance that depends on $\mathbf{x}_i$: "variance heterogeneity". And [6] accounts for CRN so the covariance matrix of e (say) $\mathbf{\Sigma_e}$ no longer equals $\sigma_e^2\mathbf{I}$ (white noise) but becomes a covariance matrix with heterogeneous variances on the main diagonal (also see [55]) and positive covariances off this diagonal. The Kriging predictor (3) then becomes

$$\widehat{y}(\mathbf{x}_{n+1}) = \mu + \mathbf{\Sigma}'_{n+1}(\mathbf{\Sigma} + \mathbf{\Sigma}_{\overline{e}})^{-1}(\overline{\mathbf{y}} - \mathbf{1}\mu) \tag{6}$$

where $\mathbf{\Sigma}_{\overline{e}}$ is the covariance matrix of $\overline{e} = \sum_{j=1}^{m_i} e_{i;j}/m_i$ and $\overline{\mathbf{y}}$ is the n-dimensional vector with the output averages $\overline{y_i} = \sum_{j=1}^{m_i} y_{i;j}/m_i$. Some

authors solve numerical problems in the computation of $\mathbf{R}^{-1}$ by assuming $\mathbf{\Sigma}_e = c\mathbf{I}$ with c being a very small number; see [36, p. 12].

As far as software for stochastic simulation with white noise is concerned, [20] provides Matlab code and [49, pp. 215-249] provides C code. Matlab code for CRN is provided on `http://www.stochastickriging.net/`. R-code is provided by [47].

3 Bootstrapping: basics

In general, bootstrapping may quantify the behavior of nonlinear statistics (such as $\widehat{\widehat{y}}(\mathbf{x})$); see the classic textbook on bootstrapping [15]. Its statistical properties such as asymptotic consistency are discussed in [7] and in the many references given in [23]. The bootstrap is a *data driven* method, so we suppose that a data set is given. For example, the data are the IID y_i $(i = 1, \ldots, n)$ (not necessarily NIID). We consider the following simple example.

Suppose the y_i are *exponentially* distributed with parameter λ: $y_i \sim Exp(\lambda)$, and we are interested in the distribution of the sample average $\overline{y}$ (a linear statistic). If $y_i \sim NIID(\mu, \sigma)$, then $\overline{y} \sim N(\mu, \sigma/\sqrt{n})$. In our example, however, $y_i \sim Exp(\lambda)$. We may then use the estimator $\widehat{\lambda} = 1/\overline{y}$. Next we can sample n new observations (say) y_i^* from $Exp(\widehat{\lambda})$: this is called *parametric bootstrapping*, which is Monte Carlo sampling with the parameter λ estimated from the data y_i; the superscript $*$ is the usual symbol denoting bootstrapped observations. From these bootstrapped observations y_i^* we compute the statistic of interest, $\overline{y^*} = \sum_{i=1}^n y_i^*/n$. To estimate the empirical density function (EDF) of $\overline{y^*}$, we repeat this resampling (say) B times, where B is called the "bootstrap sample size"; a typical value is $B = 100$. A (say) 90% confidence interval (CI) for the population mean $E(y)$ uses the (sorted from small to big) "order statistics" $\overline{y^*_{(b)}}$ with $b = 1$, ..., B; i.e., this CI is $(\overline{y^*_{(\lfloor 0.05B \rfloor)}}, \overline{y^*_{(\lceil 0.95B \rceil)}})$;see [15, pp. 170-174].

Now suppose that we do not know which type of distribution y_i has. Furthermore, n is too small for a reliable estimate of the distribution type. We can then apply *distribution-free* or *nonparametric bootstrapping*, as follows. We resample (using PRN) the n "original" observations y_i with replacement (also see Section 8). From these resampled y_i^* we compute the statistic of interest; e.g., $\overline{y^*} = \sum_{i=1}^n y_i^*/n$. Like in parametric bootstrapping, we can compute the EDF of $\overline{y^*}$ through repeating this resampling B times; this gives a CI.

Obviously, we can apply bootstrapping to estimate the EDF of more complicated statistics than the average; namely, *nonlinear* statistics. An example is the estimated skewness $\sum_{i=1}^n (y_i - \overline{y})^3/[(n-1)s^3]$ with sample

average $\overline{y}$ and sample standard deviation s. A more interesting example is the Kriging predictor with estimated parameters; see the next section.

Bootstrapping has become popular since powerful and cheap computers have become widely available. *Software* for bootstrapping is available in many statistical software packages, including the BOOT macro in SAS and the "bootstrap" command in S-Plus; see [42]. Bootstrapping is also easily implemented in Matlab, as we do in all our applications.

4 Bootstrapped variance of Kriging predictor

We expect the true predictor variance to be underestimated by $s^2(\mathbf{x})$, defined in Section 2. Therefore [14] derives a bootstrapped estimator (an alternative is the Bayesian approach in [55]). This estimator applies *parametric* bootstrapping, assuming (1). This bootstrapping first computes (say) $\widehat{\psi}' = (\widehat{\mu}, \widehat{\sigma^2}, \widehat{\theta}')$, the MLEs computed from the "original" old I/O data $(\mathbf{X}, \mathbf{y})$ where $\mathbf{X}$ is the $n \times d$ input matrix with $\mathbf{x}_i'$ $= (x_{i1}, ..., x_{id})$ and $\mathbf{y} = (y_1, ..., y_n)'$ is the corresponding output vector. To compute these MLEs, [14] uses the DACE software. These MLEs specify the distribution from which to sample *bootstrapped* observations.

To estimate the MSE of the Kriging predictor at the new $\mathbf{x}_{n+1}$, [14] samples *both* the n bootstrap outputs $\mathbf{y}^* = (y_1^*, ..., y_n^*)'$ at the old $\mathbf{X}$ *and* y_{n+1}^* at the new $\mathbf{x}_{n+1}$. These $n+1$ outputs—collected in $\mathbf{y}_{n+1}^{*\prime} =$ $(\mathbf{y}^{*\prime}, y_{n+1}^*)$—are correlated:

$$\mathbf{y}_{n+1}^* \sim N_{n+1}(\widehat{\mu}_{n+1}, \widehat{\mathbf{\Sigma}}_{(n+1)\times(n+1)}) \tag{7}$$

where $\widehat{\mu}_{n+1}$ has all its $(n+1)$ elements equal to $\widehat{\mu}$ and $\widehat{\mathbf{\Sigma}}_{(n+1)\times(n+1)}$ is the (symmetric positive semi-definite, PSD) matrix

$$\begin{bmatrix} \widehat{\mathbf{\Sigma}} & \widehat{\mathbf{\Sigma}_{n+1}} \\ \widehat{\mathbf{\Sigma}_{n+1}}' & \widehat{\sigma^2} \end{bmatrix}$$

with symbols defined below (3). The bootstrapped data $(\mathbf{X}, \mathbf{y}^*)$ resulting from (7) give the bootstrapped MLE $\widehat{\psi}^* = (\widehat{\mu}^*, \widehat{\sigma^2}^*, \widehat{\theta}^{*\prime})'$; [14] starts the search for this $\widehat{\psi}^*$ from $\widehat{\psi}$ (based on the original data $(\mathbf{X}, \mathbf{y})$). This $\widehat{\psi}^*$ gives the bootstrapped Kriging predictor $\widehat{\widehat{y}}_{n+1}^*$.

The squared errors (SEs) at the old points are zero, because classic Kriging is an exact interpolator; however, the squared error at the new point is

$$SE_{n+1} = (\widehat{\widehat{y}}_{n+1}^* - y_{n+1}^*)^2 \tag{8}$$

where y_{n+1}^* results from (7). To reduce sampling error, this bootstrapping is repeated B times (i.e., $B = 100$), which gives $\widehat{\widehat{y}}_{n+1;b}^*$ and $y_{n+1;b}^*$

with $b = 1, \ldots, B$. This gives *the* bootstrap estimator of the Kriging predictor's variance:

$$s^2(\widehat{\widehat{y}}^*_{n+1}) = \frac{\sum_{b=1}^{B} (\widehat{\widehat{y}}^*_{n+1;b} - y^*_{n+1;b})^2}{B}. \tag{9}$$

[14] gives several examples; viz., four mathematical functions in one or two dimensions, and one circuit-simulator taken from [48] with $n = 32$ observations and $d = 6$ dimensions. These examples suggest that (i) the true variance is indeed underestimated by the classic estimator, and (ii) the classic and the bootstrapped estimates do not reach their maximum at the same $\mathbf{x}$ (i.e., the bias is not an additive or multiplicative constant, but varies with $\mathbf{x}$). This second characteristic may make the bootstrapped variance estimator (9) useful in EGO.

5 EGO with bootstrapped variance

EGO is a well-known *sequential* method; i.e., it decides on its design of experiments (DOE), as experimental I/O results become available. EGO balances *local* and *global* search; i.e., it combines *exploitation* and *exploration*. Its classic reference (including predecessors of EGO) is [21]; a recent and in-depth discussion of classic EGO is [16, pp. 90-101].

Whereas classic EGO assumes deterministic simulation aimed at finding the unconstrained global minimum of the objective function, recent publications extend EGO to random simulation and constrained optimization; see [43] including [25] and also [16, pp. 125-131, 141-153], [18], [44], and [53].

We limit our survey to classic EGO, which uses the Kriging predictor $\widehat{y}$ and its classic estimated predictor variance $s^2(\mathbf{x})$ defined in Section 2. This EGO uses the following steps: (i) Find among the n *old* simulation outputs y_i $(i = 1, \ldots, n)$ the *minimum*, $\min_i y_i$. (ii) Estimate which $\mathbf{x}$ maximizes $\widehat{EI(\mathbf{x})}$, the estimated *expected improvement* (EI) compared with $\min_i y_i$ found in Step (i):

$$\max_{\mathbf{x}} \widehat{EI(\mathbf{x})} = \int_{-\infty}^{\min_i y_i} [\min_i y_i - y(\mathbf{x})] f[y(\mathbf{x})] dy(\mathbf{x}) \tag{10}$$

where $f[y(\mathbf{x})]$ denotes the distribution of the Kriging predictor for $\mathbf{x}$. EGO assumes that this distribution is Gaussian with mean $\widehat{y}(\mathbf{x})$ and variance $s^2(\mathbf{x})$. To find the *maximizer* of (10), EGO may use either a space-filling design with *candidate* points or a *global optimizer* such as the genetic algorithm (GA) in [16, p. 78] (a *local* optimizer is undesirable, because $\widehat{EI(\mathbf{x})}$ has many local optima; obviously, if $\mathbf{x} = \mathbf{x}_i$,

then $s^2(\mathbf{x}) = 0$ so $\widehat{EI(\mathbf{x})} = 0$) (iii) *Simulate* the maximizing combination found in Step (ii), *refit* the Kriging model to the old and new I/O data, and *return* to Step (i)—unless the stopping criterion is reached; e.g. $\max_{\mathbf{x}} \widehat{EI(\mathbf{x})}$ is "close" to zero.

Recently, [30] used the bootstrap estimator $s^2(\widehat{\widehat{y}}^*_{n+1})$ defined in (9) to compute the EI in (10), replacing the general distribution $f[\widehat{\widehat{y}}(\mathbf{x})]$ by $N[\widehat{\widehat{y}}_{n+1}, s^2(\widehat{\widehat{y}}^*_{n+1})]$. The procedure uses candidate points, not a GA. For all candidate points the bootstrapped predictions use the same bootstrapped MLE $\widehat{\psi}^*$, computed from the I/O data $(\mathbf{x}, \mathbf{y}^*)$.

Recently, [40] extended [30], considering—besides bootstrapped Kriging—*conditional simulation*, which gives bootstrapped outputs at old points that are exactly the same as the observed old outputs. Conditional simulation is detailed in [8, pp. 478-628]. The experimental results in [40] compare EGO with classic Kriging, bootstrapped Kriging, or conditional simulation. These results suggest that simulation analysts might wish to stick to classic EGO if they accept some possible inefficiency and prefer the simple analytical computations of classic EGO in (10)—compared with the sampling required by bootstrapped Kriging in (9) or conditional simulation.

6 Constrained optimization in random simulation

[29] derives a heuristic that is not guided by EGO, but is more related to classic operations research. This is a heuristic for constrained optimization in random simulation. The heuristic is applied to the academic (s, S) inventory system in [3] and a more practical call-center simulation in [22]. These two applications minimize one output (namely, costs), while satisfying a constraint for another output (service percentage or fill rate); moreover, the call-center simulation must satisfy a budget constraint for the deterministic inputs (namely, resources) which must be non-negative integers.

These two applications are examples of the general problem in which the output $E(y_0|\mathbf{x})$ is the objective to be minimized through the selection of $\mathbf{x} = (x_1, ..., x_d)'$, while the other $(r-1)$ outputs must satisfy prespecified threshold values c_h $(h = 1, \ldots, r-1)$, and the deterministic inputs $\mathbf{x}$ must satisfy s constraints f_g (e.g., budget constraints), and x_j must belong to the set of non-negative integers $\mathbf{N}$:

$$\begin{aligned} & Min_{\mathbf{x}} E(y_0|\mathbf{x}) \\ & E(y_h|\mathbf{x}) \geq c_h \ (h = 1, \ldots, r-1) \\ f_g(\mathbf{x}) \geq c_g \ & (g = 1, \ldots, s) \\ x_j \in \mathbf{N} \ & (j = 1, \ldots, d). \end{aligned} \tag{11}$$

To solve (11), [29] combine (i) sequentialized DOE to specify the next simulation combination (EGO also uses such DOE); (ii) Kriging to analyze the simulation I/O data (like EGO does); (iii) INLP to estimate the optimal solution from the explicit Kriging metamodels for $E(y_0)$ and $E(y_h)$.

The heuristic has modules that use free off-the-shelf software. These modules may be replaced as the knowledge in DOE, Kriging, and INLP evolves. Kriging may be replaced by other types of metamodels; e.g., radial basis functions as in [46]. Applications may have continuous inputs, so INLP may be replaced by a solver that uses the gradients, for which Kriging gives estimates "for free"; see [35] and also Section 8. Future research may adapt the heuristic for deterministic simulations with constrained multiple outputs and inputs.

Finally, [29] compares the results of this heuristic with those of the popular commercial heuristic OptQuest embedded in the Arena discrete-event simulation software; see [22]. The new heuristic requires fewer simulated input combinations and better estimated optima.

Here we discuss some salient characteristics of the heuristic. The heuristic may add a point either to improve the metamodel or to find the optimum—similar to "exploration" and "exploitation" in EGO and in several other discrete-event simulation optimization heuristics surveyed in [19]. The global Kriging metamodels should be accurate enough to enable INLP to identify clearly infeasible points (violating the constraints on the random outputs y_h) and suboptimal points (generating a too high goal output y_0). The heuristic may add points throughout the entire input-feasible area: exploration. The global Kriging metamodel for output h uses all observations for this output, obtained so far. To guide the INLP search, the heuristic simulates each point with required relative precision, to be reasonably certain of the objective values and the possible violation of the constraints; i.e., the heuristic selects the number of replications m_i such that the halfwidth of the 90% CI for the average simulation output is within 15% of the true mean for all r outputs; also see [32, pp. 500-503]. The heuristic uses CRN to improve the estimate of the optimum solution. The heuristic applies Kriging to the average output per simulated input combination, and does so for each of the r types of output. The heuristic also uses *distribution-free bootstrapping*, combined with cross-validation. This bootstrapping estimates the predictor variance for output h at the deleted combination x_i, denoted by $\widehat{var(y_h^*(\mathbf{x}_i))}$. Actually, this bootstrapping accounts for multivariate (r-variate) output; CRN (also see [26]), and non-constant m_i This bootstrap and cross-validation give the *Studentized* prediction

errors for output h of deleted combination i:

$$t_{m_i-1}^{h,i} = \frac{\overline{y_h(\mathbf{x}_i)} - y_h(-\mathbf{x}_i)}{\sqrt{\widehat{var(\overline{y_h(\mathbf{x}_i)})} + \widehat{var(y_h^*(\mathbf{x}_i))}}}$$
$$(h = 0, \ldots, r-1)\ (i = 1, \ldots, n_{cv}) \tag{12}$$

where $\widehat{var(\overline{y_h(\mathbf{x}_i)})} = \widehat{var(y_h(\mathbf{x}_i))}/m_i$ with $\widehat{var(y_h(\mathbf{x}_i))} = \sum_{r=1}^{m_i}[y_{i;h;r} - \overline{y_h(\mathbf{x}_i)}]^2/(m_i - 1)$. The highest absolute value of the $t_{m_i-1}^{h,i}$ in (12) over all r outputs and all n_{cv} cross-validated combinations gives $\max\left|t_{m_i-1}^{h,i}\right|$. Using Bonferroni's inequality implies that the traditional type-I error rate α is divided by $r \times n_{cv}$. If $\max\left|t_{m_i-1}^{h,i}\right|$ is significant, then all r Kriging models are rejected; else, the metamodels are used by INLP.

7 Taguchian robust optimization in simulation

In practice, at least some inputs of a given simulation model are *uncertain* so it may be wrong to use the optimum solution that is derived ignoring these uncertainties (as in the two preceding sections). Decision-making in such an uncertain world may use Taguchi's approach (see [52]), originally developed to help Toyota design *robust* cars; i.e., cars that perform reasonably well in many different circumstances in real life. The Taguchian approach differs from robust optimization in mathematical programming, initiated by Ben-Tal (see [4]); the latter approach is discussed in a simulation context by [54].

Taguchian robust simulation-optimization is studied in [13], replacing Taguchi's low-order polynomial metamodels by Kriging metamodels; moreover, bootstrapping is applied to quantify the variability in the estimated Kriging metamodels. Instead of Taguchi's signal/noise criterion— the ratio of the mean and the variance of the output— [13] combines Kriging with nonlinear programming (NLP) (also see Section 6). Changing the threshold values in the NLP model defined below in (13), enables the estimation of the Pareto frontier. An illustration of the resulting methodology is a deterministic economic order quantity (EOQ) inventory simulation with an uncertain input; namely, an uncertain demand rate. This example shows that robust optimization may require an order quantity that differs from the classic EOQ.

More precisely, Taguchi distinguishes between two types of factors (inputs, parameters, variables) $\mathbf{x}$: (i) *decision* (or control) factors $\mathbf{d} = (d_1, \ldots, d_k)$, which managers can control; e.g., in inventory management, the order quantity is controllable; and (ii) *environmental* (or noise) factors $\mathbf{e} = (e_1, \ldots, e_c)$, which are beyond management's control;

an example is the demand rate in inventory management. (we now use [13]'s symbols, which are not exactly the same as the symbols in the preceding sections).

Taguchi's *statistical* methods are criticized by many statisticians; see the panel discussion in [41]. Therefore [13] uses Kriging including LHS. Kriging is better in computer simulation experiments because the experimental area may be much larger than in Taguchi's real-life experiments, so in simulation a low-order polynomial may be a non-valid metamodel. LHS gives space-filling designs; references and websites for various space-filling designs are given in [23, pp. 127-130].

Whereas Taguchi focuses on the signal/noise ratio, [13] uses the following NLP model:

$$Min_{\mathbf{d}} E(w|\mathbf{d}) \text{ such that } s(w|\mathbf{d}) \leq T \tag{13}$$

where $E(w|\mathbf{d})$ is the mean of the simulation output w defined by the distribution function of the environmental variables $\mathbf{e}$; this mean is controlled through the decision factors $\mathbf{d}$; $s(w|\mathbf{d})$—or briefly s_w—is the standard deviation of the goal output w and should not exceed a given constraint value T. Unlike the variance, the standard deviation has the same scale as the mean. Next, $E(w|\mathbf{d})$ and s_w are replaced by their Kriging approximations. Obviously, the constrained minimization problem (13) is nonlinear in the decision variables $\mathbf{d}$. Decreasing T in (13) increases $E(w|\mathbf{d})$ if the constraint with the old T was binding. So, changing T gives an estimate of the Pareto-optimal efficiency frontier; i.e., $E(w|\mathbf{d})$ and s_w are criteria requiring a trade-off.

In general, simulation analysts often use LHS to obtain the I/O simulation data to which Kriging models are fitted. Such LHS is also used in [13] as part of the following two approaches, especially developed for robust optimization: (i) Similar to [12], fit two Kriging metamodels; namely, one model for the mean and one for the standard deviation—both estimated from the *simulation* I/O data. (ii) Similar to [33], fit a single Kriging metamodel to a relatively small number (say) n of combinations of $\mathbf{d}$ and $\mathbf{e}$; next use this metamodel to compute the *Kriging predictions* for the simulation output w for $N >> n$ combinations of $\mathbf{d}$ and $\mathbf{e}$ accounting for the distribution of $\mathbf{e}$.

Sub (i): Start with selecting the input combinations for the simulation model through a *crossed* (combined) design for $\mathbf{d}$ and $\mathbf{e}$—as is also traditional in Taguchian design; i.e., combine the (say) n_d combinations of $\mathbf{d}$ with the n_e combinations of $\mathbf{e}$ (an alternative would be the split-plot design in [11]). These n_d combinations are space-filling. The n_e combinations are *sampled* from the distribution of $\mathbf{e}$, using LHS for this (stratified) sampling. The resulting I/O data form an $n_d \times n_e$ matrix,

enabling estimators of the n_d conditional means and variances:

$$\overline{w_i} = \frac{\sum_{j=1}^{n_e} w_{ij}}{n_e} \text{ and } s_i^2(w) = \frac{\sum_{j=1}^{n_e}(w_{ij} - \overline{w_i})^2}{n_e - 1} \ (i = 1, \ldots, n_d). \quad (14)$$

These two estimators are unbiased, because they do not use any metamodels.

Sub (ii): Start with selecting a relatively small n (number of input combinations) using a space-filling design for the $k + c$ factors $\mathbf{d}$ and $\mathbf{e}$; i.e., $\mathbf{e}$ is not yet sampled from its distribution. Next, use these $n \times (k + c)$ simulation input data and their corresponding n outputs w to fit a Kriging metamodel for the output w. Finally, for a much larger design with N combinations, use a space-filling design for $\mathbf{d}$ but use LHS for $\mathbf{e}$ accounting for the distribution of $\mathbf{e}$. Compute the Kriging predictors $\widehat{y}$ (or $\widehat{\widehat{y}}$ in the symbols of Section 2) for the N outputs. Then derive the conditional means and standard deviations using (14) replacing n_e and n_d by N_e and N_d and replacing the simulation output w by the Kriging predictor $\widehat{y}$. Use these predictions to fit two Kriging metamodels; namely, one Kriging model for the mean output and one for the standard deviation of the output.

Sub (i) and (ii): Combining the two Kriging metamodels (for the mean and standard deviation of the simulation output) with the NLP model (13) and varying the threshold T gives the estimated Pareto frontier; this is the "original" frontier, to be distinguished from the bootstrapped frontier discussed below. The original frontier is built on estimates of the mean and standard deviation of the simulation output. To quantify the variability in the estimated mean and standard deviation, apply *distribution-free bootstrapping*. Moreover, bootstrapping assumes that the original observations are IID; however, the crossed design for $\mathbf{d}$ and $\mathbf{e}$ implies that the n_d observations on the output for a given combination of the c environmental factors $\mathbf{e}$ are not independent (this dependence may be compared with the dependence created by CRN). Therefore, the n_d-dimensional vectors $\mathbf{w}_j$ $(j = 1, \ldots, n_e)$ are resampled n_e times with replacement. This resampling gives the n_e bootstrapped observations $\mathbf{w}_j^*$. This gives the bootstrapped conditional means $\overline{w_i}^*$ and standard deviations s_i^*. To these $\overline{w_i}^*$ and s_i^* Kriging is applied. These two Kriging metamodels together with the NLP model (13) give the predicted optimal bootstrapped mean and standard deviation. Repeating this bootstrap sampling B times gives CIs. More research is necessary to discover how exactly to use these CIs to account for management's risk attitude.

Future research may address the following issues. Instead of minimizing the mean under a standard-deviation constraint as in (13), we

may minimize a specific quantile of the simulation output distribution or minimize the "conditional value at risk" (CVaR). Other risk measures are the "expected shortfall", which is popular in the actuarial literature; see [31]. Furthermore, Kriging may be replaced by "generalized linear models" (GLM) and NLP by evolutionary algorithms (EAs). The methodology may also accommodate random simulation models.

8 Convex and monotonic bootstrapped Kriging

The preceding sections showed that simulation-optimization may concern either a single or multiple outputs. In case of a single output, the analysts often assume a *convex* I/O function; see [5]. An example is the newsvendor problem in [50]. In case of multiple outputs, the analysts may minimize one output while satisfying constraints on the other outputs. An example is the call center in Section 6, in which the costs are to be minimized while the service percentage should be at least 90%. It is realistic to assume that the mean service percentage is a *monotonically* increasing function of the (costly) resources.

A major problem is that simulation models do not have *explicit* I/O functions, so all the analysts can do is run the simulation model for various input combinations and observe the resulting outputs. Next they may fit a Kriging metamodel to these observed I/O combinations. This metamodel provides an explicit approximation of the simulation model's I/O function

In this section we present both monotonicity-preserving bootstrapped Kriging metamodels and convexity-improving bootstrapped Kriging metamodels. Actually, [28] applies monotonicity-preserving bootstrapped metamodels to single-server simulation models, to improve sensitivity analysis rather than optimization. And [27] applies convexity-improving bootstrapped Kriging to various inventory simulation models to find optimal solutions. In practice, simulation analysts may indeed know that the I/O function is monotonic; e.g., as the traffic rate increases, so does the mean waiting time; as the order quantity increases, so does the mean service percentage. We assume random simulation with replications, so distribution-free bootstrapping can be applied (in deterministic simulation we could apply parametric bootstrapping.

Monotonicity-preserving bootstrapped Kriging

To obtain a monotonic Kriging metamodel, [28] assumes no CRN, and resamples—with replacement—the m_i replicated outputs $w_{i;r}$ ($r = 1, ..., m_i$) for factor combination i ($i = 1, ..., n$), and fits a Kriging metamodel to the resulting n bootstrapped averages $\overline{w_i^*}$ (this metamodel uses $\widehat{\theta}^*$). Obviously, this procedure allows variance heterogeneity of the simulation outputs so m_i is not necessarily constant. The fitted Kriging

metamodel is *accepted* only if it is monotonically increasing for all n old combinations and for a set of (say) n_c candidate combinations; the latter are selected through LHS. Monotonicity implies that the *gradients* at all these combinations are positive: $\nabla y_i^* > \mathbf{0}$ $(i = 1, \ldots, n + n_c)$; we use [28]'s symbol y rather than our symbol $\widehat{\widehat{y}}$ defined in Section 2. DACE provides estimates of all these gradients. This bootstrapped Kriging metamodel does not interpolate the original average output $\overline{w_i}$ (it does interpolate $\overline{w_i^*}$). This bootstrapping is repeated B times, but the procedure keeps only the (say) $A \leq B$ Kriging predictors with $\nabla y_{i;a}^* > \mathbf{0}$ $(a = 1, ..., A)$. For the new input combination $\mathbf{x}_{n+1}$, this gives the A predictions $y_{n+1;a}^*$.These $y_{n+1;a}^*$ give as the point estimate the sample median $y_{n+1;(\lceil 0.50A \rceil)}^*$. To obtain a (say) 90% CI, the A accepted predictions $y_{n+1;a}^*$ are sorted, which gives the order statistics $y_{(n+1;a)}^*$; these order statistics give the lower and upper CI bounds $y_{n+1;(\lfloor 0.05A \rfloor)}^*$ and $y_{n+1;(\lceil 0.95A \rceil)}^*$ If this CI turns out to be too wide, then A is increased by increasing the bootstrap sample size B. CIs in the *classic* Kriging literature assume normality of the simulation output and use the variance estimator for the Kriging predictor that ignores the random character of the Kriging parameters; see Section 4. An additional advantage of this bootstrap Kriging is that its CI obviously excludes negative values if negative values are not observed when running the simulation model.

To illustrate and evaluate this method, [28] uses the popular M/M/1 simulation model. The output is either the mean or the 90% quantile of the waiting time distribution. [28] assumes that if the analysts require monotonicity for the simulation model's I/O function, then they should obtain so many replications that the n average simulation outputs $\overline{w_i}$ also show this property: $\overline{w_i} > \overline{w_{i-1}}$. For the mean and the 90% quantile and $n = 10$ simulated traffic rates, the CI coverages turn out to be close to the nominal 90% for monotonic Kriging, whereas classic Kriging gives coverages far below the desired nominal value; for $n = 5$ the coverages of bootstrapped Kriging are still better than classic Kriging, but lower than the required 90%.

In practice, monotonicity-preserving Kriging implies sensitivity analysis that is better understood and accepted by the clients of the simulation analysts so the decision-makers trust the simulation as a decision support tool. Furthermore, estimated gradients with correct signs may improve simulation optimization, but this issue is not explored in [28].

Convexity-improving bootstrapped Kriging

[27].derives a procedure to improve the convexity of Kriging metamodels. This procedure follows the preceding monotonicity-preserving procedure. A convex function has a positive semi-definite (PSD) *Hessian* (square matrix of second-order partial derivatives). To estimate the

Hessian, [27] augments the DACE software with some extra programming; an example of the formulas for a Hessian is equation (6) in [27] for an (s, S) inventory simulation. To illustrate and evaluate this procedure, [27] use five examples, but here we discuss only the two newsvendor examples. A newsvendor must decide on the order quantity x to satisfy random demand D, for which [50] proves convexity. One example assumes that D has a uniform distribution ; the other example assumes a Gaussian distribution for D. These two examples are not discrete-event simulations, but they are random simulations. These examples do not give truly convex classic or bootstrapped Kriging metamodels. Therefore [27] *accepts* only those bootstrapped Kriging metamodels that have at least as many PSD estimated Hessians at the old plus new points, as the classic Kriging metamodel has. This convexity-improving bootstrap Kriging gives a CI for both the optimal input combination and the corresponding output.

For the input x, [27] selects $n = 10$ values because of the rule-of-thumb in [34]. To select the specific n values, LHS is used. The number of replications m_i is selected such that with 90% certainty the average simulation output per input value is within 10% of the true output; this gives $10 \leq m_i \leq 110$. Because these two examples have a single input, it is not necessary to estimate the Hessians but it suffices to check that the first-order derivatives increase—from a negative value to a positive value. The derivatives are checked at 10 old points and at 100 equally spaced points. This check shows lack of convexity in roughly half of the old and new points! (A visual check suggests that the averages do not show convexity; these averages are random.)

We expect that the accepted Kriging metamodels improve simulation-optimization. There are many simulation–optimization methods, but [27] applies a simple *grid search*; i.e., in the area of interest the Kriging predictor is computed at a grid and the combination that gives the minimum predicted output is selected. So, the A accepted Kriging metamodels give the estimated optimum outputs (say) $y^*_{a;opt}$ with $a = 1, ..., A$. The resulting order statistics $y^*_{(a);opt}$ give both a CI and the median point estimate. The same grid search can also be applied to the classic Kriging metamodel. Bootstrapped Kriging with its A accepted metamodels also gives the estimated optimum input combinations. Sorting these estimates for the optimal input gives a CI and a median. Furthermore, there is an estimated optimal input for the classic Kriging metamodel. [27] conclude that bootstrapping helps find better solutions than classic Kriging suggests; the CIs for the optimal inputs help select an experimental area for the simulation experiments in the next stage. We point out that classic Kriging does not provide a CI for the estimated optimal

input combination; it does provide a CI for the estimated output that corresponds with this input combination.

9 Conclusions

We surveyed simulation-optimization via Kriging metamodels of either deterministic or random simulation models. These metamodels may be analyzed through bootstrapping. The various sections demonstrated that the bootstrap is a versatile method, but it must be tailored to the specific problem being analyzed. Distribution-free bootstrapping applies to random simulation models, which are run several times for the same scenario. Deterministic simulation models, however, are run only once for the same scenario, so parametric bootstrapping applies assuming a Gaussian process (multivariate Gaussian distribution) with parameters estimated from the simulation I/O data.

More specifically, we focussed on the following topics. (i) EGO in deterministic simulation, using Kriging: Either parametric bootstrapping or conditional simulation gives better estimators of the Kriging predictor's variance accounting for the randomness resulting from estimating the Kriging parameters. (ii) Constrained optimization in random simulation: Distribution-free bootstrapping can help validate the Kriging metamodels, which that are combined with mathematical programming. (iii) Robust optimization accounting for an uncertain environment: Combining Kriging metamodels and mathematical programming may result in a robust solution; the effects of the randomness in the Kriging metamodels can be analyzed through distribution-free bootstrapping. (iv) Bootstrapped Kriging either preserving the monotonicity or improving the convexity of the Kriging metamodel when the simulation model is assumed to have either a monotonic or a convex I/O function.

Acknowledgement: This paper is based on a seminar I presented at the conference "Stochastic and noisy simulators" organized by the French Research Group on Stochastic Analysis Methods for COdes and NUMerical treatments called "GDR MASCOT-NUM" in Paris on 17 May 2011. I also presented a summary of this paper at the workshop "Accelerating industrial productivity via deterministic computer experiments and stochastic simulation experiments" organized at the Isaac Newton Institute for Mathematical Sciences in Cambridge on 5 - 9 September 2011. And I presented a summary at the "First Workshop On Applied Meta-Modeling", Cologne University of Applied Sciences, Gummersbach, Germany, 16 November 2012. I like to thank the coauthors with whom I wrote the various articles that are summarized in this survey.

References

[1] Ankenman, B., B. Nelson, and J. Staum (2010), Stochastic kriging for simulation metamodeling, *Operations Research*, 58, no. 2, pp. 371-382

[2] Baldi Antognini, A. and M. Zagoraiou (2010), Exact optimal designs for computer experiments via Kriging metamodeling, *Journal of Statistical Planning and Inference*, 140, pp. 2607-2617

[3] Bashyam, S. and M. C. Fu (1998), Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science*, 44, pp. 243–256

[4] Ben-Tal, A. and A. Nemirovski (2008), Selected topics in robust optimization, *Mathematical Programming*, 112, 125-158

[5] Boyd, S. and L. Vandenberghe (2004), *Convex optimization*. Cambridge University Press, Cambridge, U.K.

[6] Chen, X., B. Ankenman, and B.L. Nelson (2010), The effects of common random numbers on stochastic Kriging metamodels.Working Paper, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois

[7] Cheng, R.C.H. (2006), *Resampling methods. Handbooks in Operations Research and Management Science, Volume 13*, edited by S.G. Henderson and B.L. Nelson, pp. 415–453

[8] Chilès, J., and P. Delfiner (2012), *Geostatistics: Modeling Spatial Uncertainty. Second Edition* Wiley, New York

[9] Cressie, N.A.C. (1993), *Statistics for spatial data: revised edition.* Wiley, New York

[10] Dancik, G.M. and K.S. Dorman (2008), mlegp: statistical analysis for computer models of biological systems using R. *Bioinformatics*, 24, no. 17, pp. 1966–1967

[11] Dehlendorff, C., M. Kulahci, and K. Andersen (2011), Designing simulation experiments with controllable and uncontrollable factors for applications in health care. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 60, pp. 31-49

[12] Dellino, G., J.P.C. Kleijnen, C. Meloni. (2010), Robust optimization in simulation: Taguchi and Response Surface Methodology. *International Journal of Production Economics*, 125, pp. 52-59

[13] Dellino, G. J.P.C. Kleijnen, and C. Meloni (2012), Robust optimization in simulation: Taguchi and Krige combined. *INFORMS Journal on Computing*, 24, no. 3, pp. 471–484

[14] Den Hertog, D., J.P.C. Kleijnen, and A.Y.D. Siem (2006), The correct Kriging variance estimated by bootstrapping. *Journal Operational Research Society*, 57, pp. 400–409

[15] Efron, B. and R.J. Tibshirani (1993), *An introduction to the boot-*

strap. Chapman & Hall, New York

[16] Forrester, A., A. Sóbester, and A. Keane (2008), *Engineering design via surrogate modelling: a practical guide*. Wiley, Chichester, United Kingdom

[17] Frazier, P.I. (2011), Learning with Dynamic Programming. In: *Wiley Encyclopedia of Operations Research and Management Science*, Cochran, J.J. , Cox, L.A., Keskinocak, P., Kharoufeh, J.P., Smith, J.C. (eds.), Wiley, New York

[18] Frazier, P., W. Powell, and S. Dayanik (2009), The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21, pp. 599-613

[19] Fu, M.C. (2007), Are we there yet? The marriage between simulation & optimization. *OR/MS Today*, 34, pp.16–17

[20] Han, G. and T.J. Santner (2008), MATLAB parametric empirical Kriging (MPErK) user's guide. Department of Statistics, The Ohio State University, Columbus, OH 43210-1247

[21] Jones, D.R., M. Schonlau, and W.J. Welch (1998), Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, pp. 455-492

[22] Kelton, W.D., R.P. Sadowski, D.T. Sturrock (2007), *Simulation with Arena; 4th edition*. McGraw-Hill, Boston

[23] Kleijnen, J.P.C. (2008), *Design and analysis of simulation experiments*. Springer

[24] Kleijnen, J.P.C. (2009), Kriging metamodeling in simulation: a review. *European Journal of Operational Research*, 192, no. 3, pp. 707-716

[25] Kleijnen, J.P.C. (2013), Discussion: A discrete-event simulation perspective. *Technometrics*, 55, no. 1, pp. 21-24

[26] Kleijnen, J.P.C. and D. Deflandre (2006), Validation of regression metamodels in simulation: bootstrap approach.*European Journal of Operational Research*, 170, pp. 120–131

[27] Kleijnen, J.P.C., E. Mehdad, and W. van Beers (2012), Convex and monotonic bootstrapped Kriging. *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, pp. 543-554

[28] Kleijnen, J.P.C. and W.C.M. van Beers (2012), Monotonicity-preserving bootstrapped Kriging metamodels for expensive simulations. *Journal of the Operational Research Society*, 64, no. 5, pp. 708-717

[29] Kleijnen, J.P.C., W.C.M. Van Beers, and I. van Nieuwenhuyse (2010), Constrained optimization in simulation: a novel approach. *European Journal of Operational Research*, 202, pp. 164-174

[30] Kleijnen, J.P.C., W.C.M. van Beers, and I. van Nieuwenhuyse (2012), Expected improvement in efficient global optimization through bootstrapped Kriging. *Journal of Global Optimization*, 54, no. 1, pp. 59-73

[31] Lan, H., B.L. Nelson, and J. Staum (2010), A confidence interval procedure for expected shortfall risk measurement via two-level simulation. *Operations Research*, 58, no. 5, pp. 1481-1490

[32] Law, A.M. (2007), *Simulation modeling and analysis; fourth edition.* McGraw-Hill, Boston

[33] Lee, K.H. and G.J. Park (2006), A global robust optimization using Kriging based approximation model. *Journal of the Japanese Society of Mechanical Engineering*, 49, pp. 779-788

[34] Loeppky, J. L., Sacks, J., and Welch, W. (2009), Choosing the Sample Size of a Computer Experiment: A Practical Guide, Technometrics, 51, pp. 366-376

[35] Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002a), DACE: a Matlab Kriging toolbox, version 2.0. IMM Technical University of Denmark, Lyngby

[36] Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002b), Aspects of the Matlab toolbox DACE. IMM-Report 2002-13, IMM Technical University of Denmark, Lyngby

[37] Marrel, A., B. Iooss, S. Da Veiga, and M. Ribatet (2012), Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, 22, pp. 833-847

[38] Martin, J.D. and T.W. Simpson (2005), On the use of Kriging models to approximate deterministic computer models. *AIAA Journal*, 43, no. 4, pp. 853-863

[39] Matheron, G.F.P.M. (1975), *Random sets and integral geometry*, Wiley, New York

[40] Mehdad, E. and J.P.C. Kleijnen (2013), Bootstrapping and conditional simulation in Kriging: Better confidence intervals and optimization? CentER Discussion Paper, number CentER 2013-038, `http://arno.uvt.nl/show.cgi?fid=130596`

[41] Nair, V.N., editor.(1992), Taguchi's parameter design: a panel discussion. *Technometrics*, 34, pp. 127-161

[42] Novikov, I. and B. Oberman (2007), Optimization of large simulations using statistical software *Computational Statistics & Data Analysis*, 51, no. 5, pp. 2747-2752

[43] Picheny, V., D. Ginsbourger, Y. Richet, and G. Caplin (2013), Quantile-based optimization of noisy computer experiments with tunable precision (including Comments and Rejoinder) *Technometrics*, 55, no. 1, pp. 1-36

[44] Quan, N., J. Yin, S.H. Ng, L.H. Lee (2013), Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints. *IIE Transactions*, 45, pp. 763-780

[45] Rasmussen, C.E. and C. Williams (2006), *Gaussian processes for machine learning*, The MIT Press, Cambridge, Massachusetts

[46] Regis, R.G. (2011), Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Computers & Operations Research*, 38, pp. 837-853

[47] Roustant, O., D. Ginsbourger, and Y. Deville (2012), DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1), 1-55 (`http://www.jstatsoft.org/v51/i01/`)

[48] Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn (1989), Design and analysis of computer experiments (includes Comments and Rejoinder). *Statistical Science*, 4, no. 4, pp. 409-435

[49] Santner, T.J., B.J. Williams, and W.I. Notz (2003), *The design and analysis of computer experiments*. Springer-Verlag, New York

[50] Shapiro, A. and A. Philpott (2012), A tutorial on stochastic programming. Downloaded from `http://stoprog.org/index.html?SPTutorial/SPTutorial.html`

[51] Stein, M.L. (1999), *Statistical interpolation of spatial data: some theory for Kriging,* Springer

[52] Taguchi, G. (1987), *System of experimental designs, volumes 1 and 2.* UNIPUB/ Krauss International, White Plains, New York

[53] Villemonteix, J., E. Vazquez, and E. Walter (2009), An informational approach to the global optimization of expensive-to-evaluate functions. Journal of Global Optimization,44, no. 4, pp. 509-534

[54] Yanıkoğlu, İ., D. den Hertog, and J.P.C. Kleijnen (2013), Adjustable robust parameter design with unknown distributions. CentER Discussion Paper, 2013-022 (`http://arno.uvt.nl/show.cgi?fid=129316`)

[55] Yin, J., S.H. Ng, and K.M. Ng (2009), A study on the effects of parameter estimation on Kriging model's prediction error in stochastic simulations. *Proceedings of the 2009 Winter Simulation Conference*, edited by M.D. Rossini, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls, pp. 674-685